L7: Entry 4 of 4 File: USPT Apr 18, 2000

DOCUMENT-IDENTIFIER: US 6052681 A
TITLE: X.500 system and methods

Brief Summary Text (17):
The present application seeks to disclose a number of inventions related to the implementation of X.500 services in a RDBMS which supports SQL or any other relational language. X.500 services can be invoked via a number of protocols, such as X.500 and LDAP.

Detailed Description Text (10):
a. independence of complexity of filter--the implementation disclosed may utilise a query optimiser provided in SQL, and therefore there is no need to replicate a query optimiser in each proprietary database to which the present invention is applied,

Detailed Description Text (30):
b) the search strategy is to apply the filter over the search area using the path or parent columns, and/or;

Detailed Description Text (35):
Still furthermore, for a search, filter and subtree searches can be provided by a single pass resolution and using the path column. One invention is to utilize a `path` field to simultaneously apply an arbitrary filter over an arbitrary subtree. The complications of aliases is handled by applying the above method to a uniquely resolved subtree.

Detailed Description Text (36):
Yet another unique method is to store the "path" of each entry as a string. Each path will then be prefixed by the path of its parent entry. This is useful for the filter in the search service.

Detailed Description Text (48):
Still further, for the arbitrary filter, a dynamic SQL equivalent is built. This enables arbitrary complexity in X.500 searches.

Detailed Description Text (96):
Both the normalized data and "raw" data are stored in the database. The "raw" data is necessary so that users can retrieve the data in exactly the same format as it was originally input. As per the X.500 and LDAP standard, data received from a user, raw data, accords with ASN.1. (Abstract Syntax Notation No. 1). Thus the "Name" column in the Hierarchy Table becomes the "NameRaw" and a "NameNorm" column is added. Similarly, the "Value" column in the Object Table becomes the "ValueRaw" and a "ValueNorm" column is added.

Detailed Description Text (191):
The most powerful and useful X.500 service is the search service. The search service allows an arbitrary complex filter to be applied over a portion of the Directory Information Tree (the search area).

Detailed Description Text (192):
A filter is a combination of one or more filter items connected by the operators AND, OR and NOT. For example; surname="MASTERS" AND title="SALES MANAGER"

Detailed Description Text (194):
One technique for resolving searches is to apply the filter and then to see if any matching entries are in the search area. In this case a filter is applied to the entire tree and EID's

for all rows matching the filter are returned. Then, for each EID found, step search up through the hierarchy to see if the entry is a subordinate of the base object (i.e. the entry has a parent/grandparent/ . . . that is the base object). If the number of matches is large and the subtree small this is very inefficient. This technique doesn't cope with aliases as an alias is not a parent of the object that it points to and many aliases may point to a single object.

Detailed Description Text (195):
A second strategy is to obtain a list of all EID's in the search area and then apply the filter to these EID's. If an alias is resolved that points outside of the original search area then the subtree pointed to by the alias is expanded and the EID's in that subtree are added to the list. The filter is then applied to the set of expanded EID's. This is very poor if the search area is large.

Detailed Description Text (196):
An innovation is to simultaneously apply the filter over the search area (instead of sequentially as in the two methods described above). This is called single pass resolution. This method is considered to provide considerable performance improvement over the above methods because the rows that are retrieved are those that satisfy both the filter and scope requirements of the search.

Detailed Description Text (197):
When performing a one level search the filter is applied to all entries that have a parent equal to the EID of the base object (for example; search where parent=20 will apply the filter to entries 30, 31 and 32).

Detailed Description Text (198):
When performing a subtree search the path is used to expand the search area. The "path" of each entry is a string of numbers (e.g. "1.10.50.222." which indicates that entry 222 has a parent of 50, a grandparent of 10 and a great grandparent of 1). The path has the unique property that the path of an entry is a prefix of the path of all entries that are subordinate to the entry. That is the path of an entry forms the prefix of the paths of all entries in the subtree below the entry. Therefore when performing a subtree search we obtain the base object of the subtree and then apply the filter to all entries that have a path which is prefixed by the path of the base object (for example; to search for all entries under "Sales" we perform a search where PATH LIKE 1.11.%).

Detailed Description Text (200):
Navigate to the base object. Store the EID. In the Object Table, read nominated values from rows which match the stored EID where a filter criteria is satisfied, e.g., telephone prefix="727".

Detailed Description Text (206):
Navigate to the base object. Store the EID. Return the list of EID's which have a parent EID matching the stored EID (in Hierarchy table) and have values which satisfy the filter criteria (OBJECT table). In the Object Table, read nominated values for the returned EID's.

Detailed Description Text (215):
Navigate to the base object. Store the EID. Return the list of all EID's with a path like that of the base object (Hierarchy table) and have values which satisfy the filter criteria (OBJECT table). In the Object Table, read nominated values for the returned EID's.

Detailed Description Text (228):
Aliases are dereferenced during a search if the "search-aliases" flag in the search argument is set. The performance of the search service while dereferencing aliases becomes a two step process. Firstly, define the search area and then apply the filter to the entries within the search area. Aliases dereferenced as part of the search service can expand the search area to which the filter is applied. They also restrict the search area in that any dereferenced aliases are excluded from the search area.

Detailed Description Text (230):
If aliases are being dereferenced as part of a one level search and an alias entry is found then the alias must be resolved (using the Object table or the A.sub.-- EID). The aliased object is then added to the search area to which the filter is applied. In a oneLevel search

where aliases are found the search area will consist of non-alias entries directly subordinate to the base object and all dereferenced aliases.

Detailed Description Text (264):
The Search Table is used to resolve filters in the Search service. It is also used to find values during Compare, Modify and ModifyRDN. The Search table contains one row for each attribute value of each entry. Only the normalised values are stored in this table.

Detailed Description Text (277):
Search/Subtree--for finding EIDs that match a filter over a whole subtree (where the base object is not the root) (TREE joined with SEARCH).

Detailed Description Text (278):
Search/OneLevel--for finding EIDs that match a filter one-level under the base object (DIT joined with SEARCH).

Detailed Description Text (374):
5.5 Search Service

Detailed Description Text (375):
The Search Service is the most complex of all X.500 services. Search arguments indicate where to start the search (baseObject), the scope of the search (subset), the conditions to apply (filter) and what information should be returned (selection). In addition, a flag is passed to indicate whether aliases should be dereferenced (searchAliases).

Detailed Description Text (376):
The possible values for subset are baseObject, oneLevel and wholeSubtree. Base object indicates that the search filter will only be applied to attributes and values within the base object. OneLevel indicates the Search filter will be applied to the immediate subordinates of the base object. Whole subtree indicates the Search filter will be applied to the base object and all of its subordinates.

Detailed Description Text (377):
A simple example of a filter condition would be: surname="EVANS" or telephoneNumber PRESENT.

Detailed Description Text (381):
Apply the filter to attributes and values in the Search Table with the EID of the selected object.

Detailed Description Text (382):
If the filter condition is matched, return the Entry Information from the Entry Table.

Detailed Description Text (387):
Using the Search and DIT Tables, apply the filter (attribute/value conditions) and the scope (PARENT=EID of selected object and any aliases dereferenced). A list of matching EID's will be returned.

Detailed Description Text (395):
Using the Search and Tree Tables, apply the filter (attribute/value conditions) and the scope (PATH LIKE PATH prefix of the selected object) to each unique base object. A list of matching EID's will be returned.

Detailed Description Text (402):
a Filter of "surname, substring initial=M". (Look for all surnames beginning with "M")

Detailed Description Text (410):
Apply the filter and scope simultaneously. i.e. Using the Search Table, obtain a list of EID's from the target list where AID=4 and the value begins with "M" joined with the Tree Table who's PATH is LIKE `1.11.%`. The matching EID's are 30 and 31.

Detailed Description Text (498):
As the format of most services is known, many instances of these services can be resolved using static SQL statements. More complex services, such as searches with complex filters, can be

resolved using dynamic SQL. This enables arbitrarily complex searches to be performed.

Detailed Description Paragraph Table (25):

TABLE 4.1 Basic column usage X.500 Value Value Par- Name Name Service Table EID AID VID Norm Raw ent Alias Norm Raw Path Navigate H R S R S R Read O S (S)/R R R R R Compare O S S S List H S R R Search- O S/R S (S) (S) (S) filter Search- S/R (S)/R R R R R result Add H/O S Remove H/O S Modify O S S S S Modify RDN H/O S S S S

Detailed Description Paragraph Table (42):

X.500 definition

Argument Description baseObject The Distinguished Name of the baseObject subset baseObject, oneLevel or wholeSubtree filter search conditions searchAliases a flag to indicate whether aliases among subordinates of the base object should be dereferenced during the search. selection EIS as for READ. The attributes and values to be returned. Common Arguments Result Description DistinguishedName The DN of the selected object (returned if an alias is dereferenced) entries Attributes & values (as defined in selection) for the entries which satisfy the filter. partialOutcomeQualifier An indication that an incomplete result was returned, eg, a time limit or size limit restriction. Common Results

Detailed Description Paragraph Table (54):

TABLE 7A Service Database Size (number of entries) Operation Qualifier Detail 1K 10K 20K 50K 100K 200K BIND anonymous 0.00 0.00 0.00 0.00 0.00 0.00 LIST level 1 4 items 0.05 0.05 0.05 0.05 0.05 0.05 level 3 4 items 0.06 0.06 0.06 0.06 0.06 0.06 level 4 1 00 items 0.22 0.23 0.23 0.24 0.23 0.24 READ level 4 1 item, all info 0.07 0.07 0.07 0.07 0.07 0.08 level 4 (via alias) 1 item, all info 0.07 0.07 0.07 0.07 0.07 0.07 SEARCH 1 level, equality 100 entries, 1 item 0.12 0.12 0.12 0.12 0.13 0.13 1 level, initial 100 entries, i item 0.13 0.14 0.15 0.15 0.15 0.14 1 level, any 100 entries, 1 item 0.30 0.35 0.33 0.32 0.36 0.29 1 level, final 100 entries, 1 item 0.24 0.35 0.31 0.30 0.35 0.28 subtree, equality 1K, 1 item, level 1 0.11 0.11 0.11 0.11 0.11 0.11 10K, 1 item, level 1 xxx xxx 0.12 0.12 0.12 0.12 20K, 1 item, level 1 xxx xxx xxx 0.12 0.13 0.12 50K, 1 item, level 1 xxx xxx xxx xxx 0.13 0.13 100K, 1 item, level 1 xxx xxx xxx xxx xxx 0.12 subtree, initial 1K, 1 item, level 1 0.13 0.12 0.12 0.12 0.12 0.11 10K, 1 item, level 1 xxx xxx 0.11 0.12 0.12 0.12 20K, 1 item, level 1 xxx xxx xxx 0.13 0.12 0.12 50K, 1 item, level 1 xxx xxx xxx xxx 0.13 0.12 100K, 1 item, level 1 xxx xxx xxx xxx xxx 0.11 full, complex OR all entries, 1 item 0.09 0.09 0.09 0.09 0.09 0.09 full, complex AND all entries, 1 item 0.11 0.11 0.11 0.11 0.11 0.11 full, complex OR/AND all entries, 1 item 0.26 0.28 0.29 0.28 0.29 0.26 full, complex AND/OR all entries, 1 item 0.12 0.12 0.13 0.14 0.13 0.12 full, complex AND/AND all entries, 1 item 0.16 0.15 0.16 0.17 0.18 0.18 full, complex all entries, 1 item 0.18 0.18 0.18 0.19 0.20 0.26 AND/AND/AND full, equality all entries, 1 item 0.08 0.08 0.08 0.08 0.08 0.08 full, no filter, all-info all entries, 10 items 0.30 0.74 0.43 0.59 0.49 0.67 full, no filter, all-info all entries, 100 items 1.36 1.84 1.50 1.79 1.82 1.86 full, initial all entries, 1 item 0.08 0.08 0.08 0.08 0.08 0.08 ADD level 5 100 sisters 0.22 0.19 0.22 0.20 0.19 0.19 MODIFY level 5 100 sisters 0.09 0.11 0.11 0.11 0.11 0.11 RENAME level 5 100 sisters 0.15 0.16 0.15 0.16 0.16 0.15 DELETE level 5 100 sisters 0.17 0.16 0.17 0.17 0.17 0.19 UNBIND 0.00 0.00 0.00 0.00 0.00 0.00 Notes: 1. All searches and reads return all info 2. All tests were performed under the following environment: * Sun SparcStation 5 with 32Mb of memory (entry level UNIX machine) * Ingres 6.4/04 configured for 32 users (standard Ingres installation) * DSA prototype V2.1.2 * Timings measured at DSA console (ie does not include network overheads) All numbers are in units of seconds and "K" means 1,000's.

Current US Original Classification (1):
707/3

Current US Cross Reference Classification (1):
707/4

CLAIMS:

9. An application as claimed in any one of claims 4, 5, 6, 7 or 8, where the X.500 services are invoked using at least one of an X.500 and LDAP protocol.

12. A database as claimed in claim 11, where the X.500 services are invoked using at least one of an X.500 and LDAP protocol.

17. An apparatus as claimed in any one of claims 1, 2, 3, 14, 15 or 16, where the X.500 services are invoked using an X.500 and/or LDAP protocol.

18. A method of searching an area in a database using search services, the method comprising the step of:

applying a filter limited to a selected area of a directory information tree for said database.

19. A method as claimed in claim 18, where the step of applying a filter is applied only to syntax-normalized meta data.

20. A method of searching an area in a database which includes aliases, the method comprising the steps of:

expanding the search area by resolving aliases until a set of search areas with no unresolved aliases is found; and

applying a filter limited to a selected area of a directory information tree for said database to said set of search areas.

22. A method as claimed in claim 18, 19 or 20, wherein evaluating the filter and the scope is performed by single pass resolution.

30. A method as claimed in any one of claims 13, 14, 21, 23, 24, 25, 26, 27, 28 and 29, where the X.500 services are invoked using at least one of an X.500 and LDAP protocol.

Previous Doc Next Doc Go to Doc#